

Tuning systemd for Embedded

Alison Chaiken

alison_chaiken@mentor.com

Mar. 23, 2015

Latest version: http://she-devel.com/ELC_systemd.pdf

Auxiliary files: http://she-devel.com/ELC_auxiliary.tar.bz2



Text in [blue](#) is hyperlinked.

?



Quiz:



what is the most widely
used

Linux init system?



Linux needs to keep innovating

“No one has a guaranteed position in the technology industry.” -- Bill Gates, *Pirates of Silicon Valley*

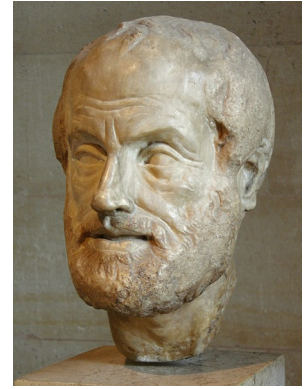
“The only thing that can ever hurt Linux is Linux itself.” -- GKH, *Linux Action Show*

“Success is a self-correcting phenomenon.” -- Gary Hamel



Licensed under CC BY-SA 3.0
<http://commons.wikimedia.org/wiki/File:Fire-lite-bg-10.jpg#mediaviewer/File:Fire-lite-bg-10.jpg>

Philosophy of systemd



Extract duplicate functionality from daemons and move it to systemd core or kernel.

Replace /etc scripts with declarative config files.

Expose newer kernel APIs to userspace via a simple interface.

systemd is:

- ***modular***;
- ***asynchronous*** and ***concurrent***;
- described by ***declarative*** sets of properties;
- bundled with analysis tools and ***tests***;
- features a fully ***language-agnostic*** API.

One daemon to rule them all

xinetd: a daemon to lazily launch **internet services** when activity is detected on an AF_INET socket

systemd: a daemon to lazily launch **any system service** when activity is detected on an AF_UNIX socket (oversimplification)

Complexity arising from many similar small units



init.d scripts \Rightarrow systemd units

- Unit's action and parameters: ExecStart=
- Dependencies: Before=, After=, Requires=, Conflicts= and Wants=.
- Default dependencies:
 - Requires= and After= on basic.target;
 - Conflicts= and Before= on shutdown.target.
- Types of unit files: **service**, socket, device, mount, scope, slice, automount, swap, **target**, path, timer, snapshot

sysVinit runlevels \approx systemd targets

- Targets are *synchronization points*.
- Check `/lib/systemd/system/runlevel?.target` symlinks:

multi-user.target (runlevel 3 == text session)

graphical.target (runlevel 5 == graphical session)

- **Select boot-target :**
 - via `/etc/systemd/system/default.target` symlink;
 - appending number or `systemd.unit=<target>` to bootargs.



plus: intuitively exposes kernel interfaces

- Including Capabilities, Watchdog, Cgroups and kdbus ('coming attraction')
- Kernel features configurable via simple ASCII options in unit files.
- Encourages creation of system ***policies*** via unit templates.

systemd and cgroups

- cgroups are a kernel-level mechanism for allocating resources: storage, memory, CPU and network.
- *slices* are groups of *daemons* whose resources are managed jointly.
- systemd *scopes* are resultant groups of *user* processes.
- Can set BlockIOWeight, IOSchedulingPriority, OOMScoreAdjust, CPUShares, MemoryLimit ...

Demo Example: limiting memory usage of Firefox.

systemd and security: granular encapsulation via kernel's *capabilities*

- CapabilityBoundingSet
- PrivateTmp, PrivateDevices, PrivateNetwork
- JoinNamespaces
- ProtectSystem (/usr and /etc), ProtectHome
- ReadOnlyDirectories, InaccessibleDirectories
- systemd-nspawn: systemd's native containers

Demo Example: limiting privileges of root-initiated program

systemd and watchdogs

- Support for soft or hard watchdogs
- RuntimeWatchdogSec sets a timer for petting the dog
- ShutdownWatchdogSec sets a timer to force reboot if shutdown hangs

Demo Example: systemd and softdog

resource utilization

- systemd-211 in Poky includes 17 packages = 8 MB.
- systemd-219 builds 90 MB of executables (not all needed).
- **minimal build** = systemd, udevd and journald.
- Memory (RSS) of fully featured build: ≈ 9 MB; minimum build ≈ 5 MB.
- Features added/removed via './configure'.
- Get rid of D-Bus, syslog and bash?

using the systemd journal



- Easily pushed to a remote.
- Can be cryptographically 'sealed'.
- Configurable max size and rotation.
- Log-reading tools are simple:

```
journalctl -xn
```

```
journalctl -p err
```

```
journalctl -u cron
```

```
journalctl -o json-pretty
```

```
systemctl status
```

```
systemctl is-failed bluetooth
```

```
systemctl --failed
```


Other embedded-relevant features

- Support for read-only rootfs
- Remote journaling via HTTP requests
- Coredumpctl:
 - read last core file
 - read stack trace automatically written to journal
- Timesyncd: lightweight network-time daemon
- 'systemctl snapshot' captures state to which the system can be restored

systemd prevents self-injury!

- Test out new units by trying them:

- in /run
- in *.conf.d directory
- via bootargs



- No need ever to modify files in /lib/systemd.
- Services linked into basic.target.wants (\approx runlevel 1) that won't work until graphical.target (runlevel 5) will start properly if their dependencies are correctly stated.

systemd and backwards compatibility



system updates

Ye Good Olde Days:

- update kernel and modules
- separately update root fs

Newfangled:

- update kernel and modules
- [update device-tree](#)
- separately update root fs

stability of systemd


Old:

- update kernel and modules
- separately update root fs

New:

- update kernel and modules
- **update device-tree?**
- separately update root fs

Newer:

- update kernel and modules
- update device-tree?
- *update systemd?* 
- separately update root fs

[systemd-devel] [PATCH] Drop the udev firmware loader

Lennart Poettering [lennart at poettering.net](mailto:lennart@poettering.net)

Sat May 31 22:45:17 PDT 2014

To make this clear, we expect that systemd and kernels are updated in lockstep. We explicitly do not support really old kernels with really new systemd. So far we had the focus to support up to 2y old kernels (which means 3.4 right now), but even that should be taken with a grain of salt, as we already made clear that soon after kdbus is merged into the kernel we'll probably make a hard requirement on it from the systemd side.

systemd's 'Interface Portability and Stability Chart'

API	Type	Covered by Interface Stability Promise	Fully documented	Known External Consumers
hostnamed	D-Bus	yes	yes	GNOME
localed	D-Bus	yes	yes	GNOME
timedated	D-Bus	yes	yes	GNOME
initrd interface	Environment, flag files	yes	yes	dracut, ArchLinux
Container interface	Environment, Mounts	yes	yes	libvirt/LXC
Boot Loader interface	EFI variables	yes	yes	gummiboot
Service bus API	D-Bus	yes	yes	system- config- services
logind	D-Bus	yes	yes	GNOME
sd-login.h API	C Library	yes	yes	GNOME, PolicyKit , ...
sd-daemon.h API	C Library or Drop-in	yes	yes	numerous
sd-id128.h API	C Library	yes	yes	-
sd-journal.h API	C Library	yes	yes	-
sd-readahead.h API	C Drop-in	yes	yes	-



Deprecated!

developing systemd

- git clone [git://anongit.freedesktop.org/systemd/systemd](https://anongit.freedesktop.org/systemd/systemd)
- systemd-devel list: submit patches or ask questions
- Featureful utility library in *src/shared/*
#define streq(a,b) (strcmp((a),(b)) == 0)
#define strneq(a, b, n) (strncmp((a), (b), (n)) == 0)
#define strcaseeq(a,b) (strcasecmp((a),(b)) == 0)
#define strncaseeq(a, b, n) (strncasecmp((a), (b), (n)) == 0)
- Complex but automated build system with many dependencies.
- 'Plumbing' dev tools in */lib/systemd*, 'porcelain' tools in */bin*
find /lib/systemd -executable -type f

Summary

- Systemd has:
 - a superior design;
 - tight integration with the Linux kernel;
 - a vibrant developer community.
- systemd is the less stable part of kernel's ABI.
- Mostly things will 'just work'.
- systemd exemplifies modernization Linux needs to stay competitive.



photo
courtesy
Jym
Dyer

Thanks

- Vladimir Pantelic, Tom Gundersen, Lennart Poettering, Jeff Waugh, Ivan Shapovalov, Mantas Mikulėnas, Stephanie Lockwood-Childs and Jon Stanley for corrections and advice.
- [Bill Ward](#), [Jym Dyer](#) and Janet Lafleur for use of their images.



Resources

- Man pages are part of [systemd git](#) repo.
- freedesktop.org: systemd [mailing list archives](#) and [wiki](#)
- Poettering's [Opointer.de](#) blog
- 🏠➡️ At [wayback machine](#): “Booting up” articles
- [Neil Brown series](#) at LWN
- 🏠➡️ Fedora's [SysVinit to systemd cheatsheet](#)
- Poettering's '[What's new](#)' talk from FOSDEM 2015
- Josh Triplett's [Debconf talk video](#)
- Linux Action Show interviews with [Mark Shuttleworth](#) and [Lennart Poettering](#)

Leftover Materials

Understanding dependencies

Try:

```
systemctl list-dependencies basic.target
```

```
systemctl list-dependencies –after tmp.mount
```


Understanding dependencies, p. 2

Try:

```
systemd-analyze dot rescue.target
```

```
systemd-analyze dot basic.target > basic.dot
```

```
dot -Tsvg basic.dot -o basic.svg
```

```
eog basic.svg (or view basic.svg with any web browser)
```

SysV already has a big service manager: bash

```
[user@localhost]$ ls -l /sbin/init
```

26k

```
[user@localhost]$ ls -lh /bin/bash
```

1008K

```
[user@localhost]$ ls -lh /lib/systemd/systemd
```

1.3M



Greg K-H: “Tightly-coupled components”

There are a number of folk in the Linux ecosystem pushing for a small core of tightly coupled components to make the core of a modern linux distro. The idea is that this “core distro” can evolve in sync with the kernel, and generally move fast. This is both good for the overall platform and very hard to implement for the “universal” distros.

Martin Langhoff

🕒 2 years ago 🔁

+ Share

Originally from <https://lwn.net/Articles/494095/>

[Socket activation demo with cups and ncat]

systemd and udev

- udev is a kernel facility that handles device events.
 - merged into the systemd project.
- Rules are enabled by placement in `/lib/udev/rules.d`, unlike systemd unit enablement.
- Rule loading is ordered by numeric filename prefix, like old sysVinit scripts.

udev is still old-school

Try:

```
ls /lib/udev/rules.d
```

```
cat /lib/udev/rules.d/99-systemd.rules
```

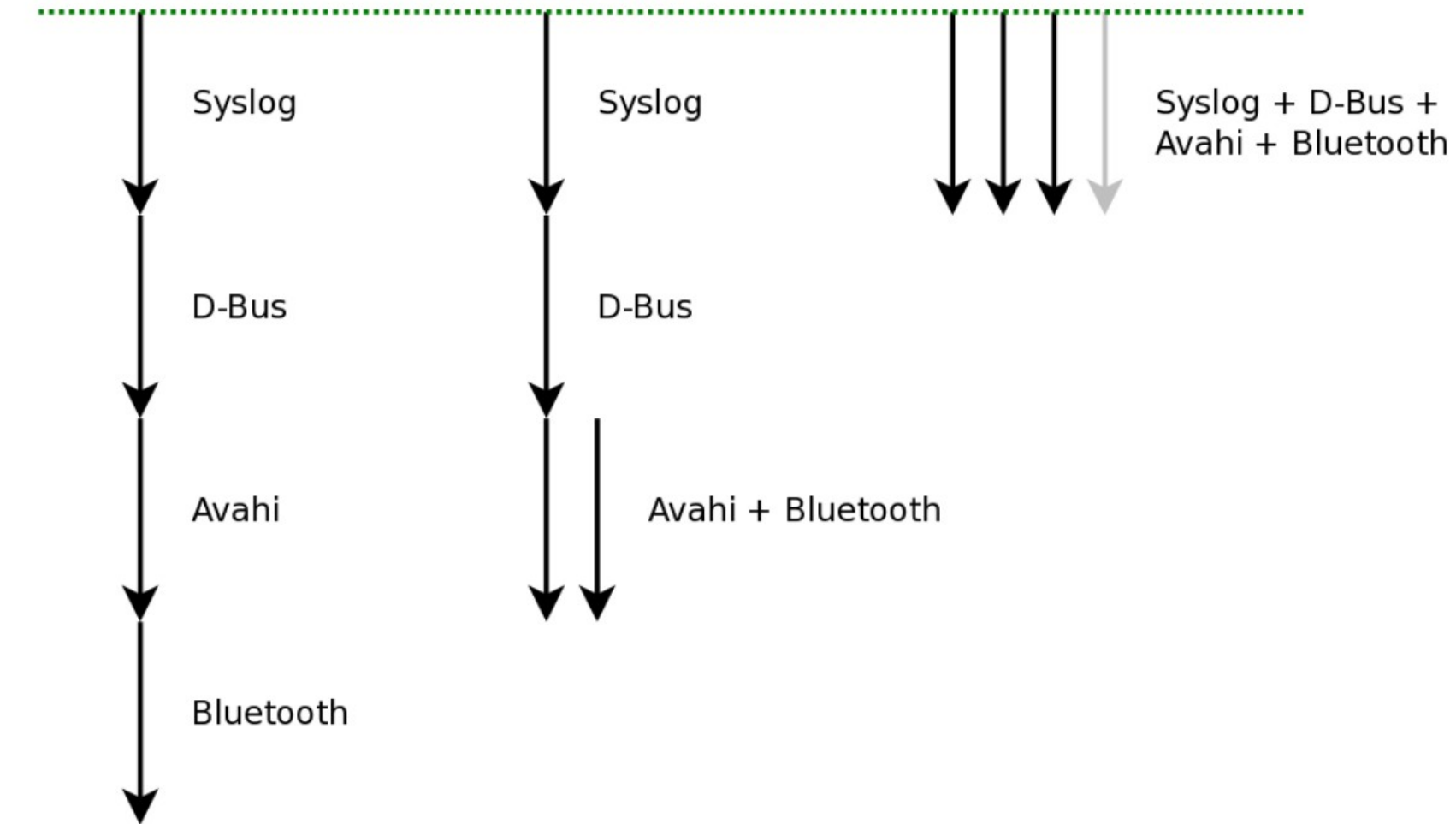

Hierarchy of unit files for system and user sessions

- Organized into *system* and *user* units
- */lib/systemd/system*: systemd upstream defaults for system-wide services
- */etc/systemd/system*: local customizations by *override* and *extension*
- */lib/systemd/user*: systemd's upstream defaults for per-user services
- *\$HOME/.local/share/systemd/user* for user-installed units
- 'drop-ins' are run-time extensions

Serial

Linked list

Fully parallel



Traditional SysV

Suse/Ubuntu ~~Parallelization~~

systemd

Upstart

photo
courtesy
Bill
Ward



Modularity can produce complexity

systemd and outside projects: [CoreOS](#)

- **networkd** was initially contributed by CoreOS developers.
- CoreOS's **fleet** “tool that presents your entire cluster as a single init system” is based on systemd.
 - Spin up new containers due to events on sockets.
- CoreOS devs are outside systemd inner circle.
- systemd has many patches from Arch, Intel, Debian . . .



systemd in embedded systems

- systemd is widely adopted in embedded systems because
 - proper allocation of resources is critical;
 - fastboot is required;
 - customization of boot sequence is common.
- Lack of backward compatibility for older kernels (due to firmware loading) is a pain point.
- Embedded use cases are not always understood by systemd devs.

Try: 'systemctl isolate multi-user.target'
[warning: **KILLS X11**]

[runlevel demo with Fedora Qemu and Firefox]

systemd is easy to use

- systemd utilities:
 - *Try: `apropos systemd | grep ctl`*
- All-ASCII configuration files: no hidden “registry”.
- Customization is by **overriding** default files.
- Many choices are controllable via symlinks.
- Bash-completion by default.
- Backwards compatibility with SysVinit

Override your defaults!

- *Replace* a unit in **/lib** (upstream) by creating one of the same name in **/etc** (local changes).
- *Add* services to boot by symlinking them into `/etc/systemd/system/default.target.wants.`
- 'mask' unit with link to `/dev/null`.
- *Best practice*: do not change the files in `/lib/systemd`.
- Read in-use unit with `'systemctl cat'`.

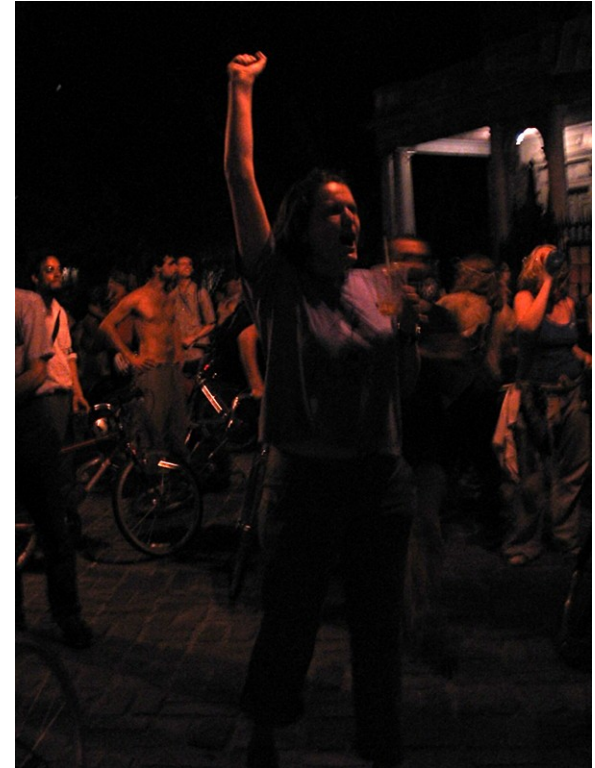


photo courtesy
Jym Dyer

Extensions: drop-ins

Try: systemd-delta

Try: systemctl cat <list from 1st command>

Old way	New way	History
X11 manages graphics memory	Kernel's drm manages graphics memory	"Linux Graphics Drivers: an Introduction," p. 26
static /dev, then devfs	udev	
getrlimit, setrlimit	cgroups	
KDE3 and GNOME2	KDE4 and GNOME3	KDE and GNOME
sysVinit	systemd	in progress
X11 client-server model	Wayland compositor	

Crux of the problem: [Dave Neary](#)

“There is no freedesktop.org process for proposing standards, identifying those which are proposals and those which are de facto implemented, and perhaps more importantly, there is no process for building consensus around a specification . . .”

(comment regarding GNOME3)

systemd is . . .

- the basis of Fedora, RHEL, CentOS, OpenSUSE, Ubuntu, Debian and much embedded.
- **praised** by Jordan Hubbard of FreeBSD.
- tightly integrated with Linux kernel cgroups.
- the reference implementation for udev and for kdbus userspace access.

Customizing your installation

- *Replace* a unit in `/lib` (upstream) by creating one of the same name in `/etc` (local changes).
- *Add* services to boot by symlinking them into `/etc/systemd/system/default.target.wants`.
- *Best practice*: do not change the files in `/lib/systemd`

Sequence of targets on a typical system

>\$ ls -l /lib/systemd/system/default.target

/lib/systemd/system/default.target -> graphical.target

>\$ cat /lib/systemd/system/graphical.target

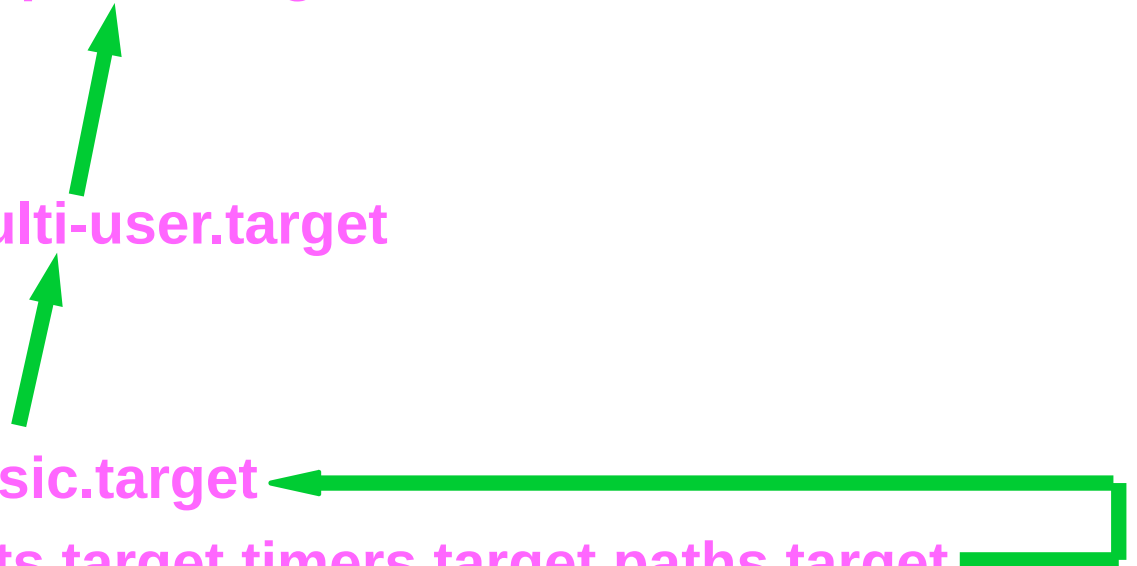
After=multi-user.target

>\$ cat /lib/systemd/system/multi-user.target

After=basic.target

>\$ cat /lib/systemd/system/basic.target

After=sysinit.target sockets.target timers.target paths.target
slices.target



Example: set display manager

```
[user@localhost ~]$ ls -l `locate display-manager.service`
```

```
lrwxrwxrwx. 1 root root 35 Dec 11 2013
```

```
/etc/systemd/system/display-manager.service ->
```

```
/usr/lib/systemd/system/gdm.service
```

```
[user@localhost ~]$ cat /usr/lib/systemd/system/gdm.service
```

```
[Unit]
```

```
Description=GNOME Display Manager
```

```
[...]
```

```
[Install]
```

```
Alias=display-manager.service
```

or

```
WantedBy=graphical.target
```

sysinit, sockets and multi-user are composite targets

>\$ ls /lib/systemd/system/**multi-user.target**.wants/

dbus.service@ systemd-ask-password-wall.path@ systemd-
update-utmp-runlevel.service@ getty.target@

>\$ ls /lib/systemd/system/**sockets.target**.wants:

dbus.socket@ systemd-shutdown.socket@
systemd-initctl.socket@ systemd-udev-control.socket@

>\$ ls /lib/systemd/system/**sysinit.target**.wants:

cryptsetup.target@ systemd-journald.service@
debian-fixup.service@ systemd-journal-flush.service@

Symlinks replace lines of conditional code in SysVinit scripts.

Example: change the default target

```
[alison@localhost ~]$ ls /etc/systemd/system/default.target
/etc/systemd/system/default.target ->
    /lib/systemd/system/graphical.target
```

```
[alison@localhost ~]$ sudo rm /etc/systemd/system/default.target
[alison@localhost ~]$ sudo ln -s /lib/systemd/system/multi-user.target
    /etc/systemd/system/default.target
```

```
[alison@localhost ~]$ ~/bin/systemd-delta
[ . . . ]
[REDIRECTED] /etc/systemd/system/default.target →
    /usr/lib/systemd/system/default.target
```

problems

- systemd *is* modular, but:
 - interoperability with other SW may be inadequately tested.
- Potentially rocky piecemeal transition by distros.
 - e.g., Debian installer doesn't warn about a separate /usr partition.
- Merciless deprecation of features (firmware loading, readahead . . .).
- Frequent releases, not particularly stable.

Taxonomy of systemd dependencies

Requires, RequiresOverridable, Requisite, RequisiteOverridable,
Wants, BindsTo, PartOf, Conflicts, Before, After, OnFailure
PropagateReloadsTo, ReloadPropagateFrom,

Brandon Philips at linux.conf.au

kernel
systemd
etcd
ssh
docker

python
java
nginx
mysql
openssl

o distro distro distro distro distro

app

Brandon Philips at linux.conf.au

kernel
systemd
etc
ssh
docker

o distro distro distro distro distro

python
java
nginx
mysql
openssl

app

Design

